


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY

Feedback

(stack and unwind)

Published before December 2002

Found 2

Terms used: [stack](#) [unwind](#)Sort results by
☒ [Save results to a Binder](#)

 Refine these results with [Advanced :](#)
 Try this search in [The ACM Guide](#)
Display results
☐ [Open results in a new window](#)

Results 1 - 20 of 20

 1 [Generational stack collection and profile-driven pretenuring](#)

Ads by C



Perry Cheng, Robert Harper, Peter Lee

May 1998 PLDI '98: Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation

Publisher: ACM

Full text available: pdf(1.56 MB)

[Additional Information: full citation, abstract, references, cited by, index terms](#)

Bibliometrics: Downloads (6 Weeks): 2, Downloads (12 Months): 18, Citation Count: 32

This paper presents two techniques for improving garbage collection performance: generational stack collection and profile-driven pretenuring. The first is applicable to stack-based implementations of functional languages while the second is useful for ...

[Online Full-text at Queue Library.](#)
[www.Queue](#)
[Science Read h inspire enthusi science](#)
[www.intel](#)

 2 [Generational stack collection and profile-driven pretenuring](#)


Perry Cheng, Robert Harper, Peter Lee

May 1998 ACM SIGPLAN Notices, Volume 33 Issue 5

Publisher: ACM

Full text available: pdf(1.56 MB)

[Additional Information: full citation, abstract, references, cited by, index terms](#)

Bibliometrics: Downloads (6 Weeks): 2, Downloads (12 Months): 18, Citation Count: 32

This paper presents two techniques for improving garbage collection performance: generational stack collection and profile-driven pretenuring. The first is applicable to stack-based implementations of functional languages while the second is useful for ...

[Educational Techno Want to Classro Educati Latest Techno](#)
[www.Full](#)

 3 [Support for garbage collection at every instruction in a Java compiler](#)


James M. Stichnoth, Guei-Yuan Lueh, Michał Cierniak

May 1999 ACM SIGPLAN Notices, Volume 34 Issue 5

Publisher: ACM

Full text available: pdf(1.06 MB)

[Additional Information: full citation, abstract, references, cited by, index terms](#)

Bibliometrics: Downloads (6 Weeks): 8, Downloads (12 Months): 52, Citation Count: 12

A high-performance implementation of a Java Virtual Machine¹ requires a compiler to translate Java bytecodes into native instructions, as well as an advanced garbage collector (e.g., copying or generational). When the Java heap is exhausted ...

[BlackB Univer: Educati Connec BlackB Out Ho](#)
[www.Black](#)

Keywords: Java, compilers, garbage collection

 4 [A study of exception handling and its dynamic optimization in Java](#)

Takeshi Ogasawara, Hideaki Komatsu, Toshio Nakatani



November 2001 ACM SIGPLAN Notices, Volume 36 Issue 11

Publisher: ACM

Full text available: pdf(190.18 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 14, Downloads (12 Months): 101, Citation Count: 18

Optimizing exception handling is critical for programs that frequently throw exceptions. We observed that there are many such exception-intensive programs in various categories of Java programs. There are two commonly used exception handling techniques, ...

5 [A hybrid execution model for fine-grained languages on distributed memory multicomputers](#)



John Plevyak, Vijay Karamcheti, Xingbin Zhang, Andrew A. Chien

December 1995 Supercomputing '95: Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Volume 00, Volume 00

Publisher: ACM

Full text available: html(59.68 KB) ps(419.70 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 2, Downloads (12 Months): 9, Citation Count: 6

While fine-grained concurrent languages can naturally capture concurrency in many irregular and dynamic problems, their flexibility has generally resulted in poor execution efficiency. In such languages the computation consists of many small threads which ...

6 [A study of exception handling and its dynamic optimization in Java](#)



Takeshi Ogasawara, Hideaki Komatsu, Toshio Nakatani

October 2001 OOPSLA '01: Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications

Publisher: ACM

Full text available: pdf(190.18 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 14, Downloads (12 Months): 101, Citation Count: 18

Optimizing exception handling is critical for programs that frequently throw exceptions. We observed that there are many such exception-intensive programs in various categories of Java programs. There are two commonly used exception handling techniques, ...

7 [Support for garbage collection at every instruction in a Java compiler](#)



James M. Stichnoth, Guei-Yuan Lueh, Michał Cierniak

May 1999 PLDI '99: Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation

Publisher: ACM

Full text available: pdf(1.06 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 8, Downloads (12 Months): 52, Citation Count: 12

A high-performance implementation of a Java Virtual Machine¹ requires a compiler to translate Java bytecodes into native instructions, as well as an advanced garbage collector (e.g., copying or generational). When the Java heap is exhausted ...

Keywords: Java, compilers, garbage collection

8 [Defined functions](#)



December 1983 ACM SIGAPL APL Quote Quad, Volume 14 Issue 2

Publisher: ACM

Full text available: pdf(631.48 KB)

Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 0, Downloads (12 Months): 4, Citation Count: 0

Algorithms written in APL are called defined functions. A defined function consists of a header line and zero or more body lines. The header line indicates the name and syntax class of the function and gives a list of names to be localised and, in the ...

9 [There and back again](#)



Olivier Danvy, Mayer Goldberg

October 2002 | CFP '02: Proceedings of the seventh ACM SIGPLAN international conference on Functional programming

Publisher: ACM

Full text available: pdf(119.96 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 1, Downloads (12 Months): 19, Citation Count: 1

We present a programming pattern where a recursive function traverses a data structure---typically a list---at return time. The idea is that the recursive calls get us there (typically to a base case) and the returns get us back again *while traversing* ...

Keywords: Catalan numbers, Symbolic convolutions, continuation-based programming, palindrome detection

10 [There and back again](#)



Olivier Danvy, Mayer Goldberg

September 2002 ACM SIGPLAN Notices, Volume 37 Issue 9

Publisher: ACM

Full text available: pdf(119.96 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 1, Downloads (12 Months): 19, Citation Count: 1

We present a programming pattern where a recursive function traverses a data structure---typically a list---at return time. The idea is that the recursive calls get us there (typically to a base case) and the returns get us back again *while traversing* ...

Keywords: Catalan numbers, Symbolic convolutions, continuation-based programming, palindrome detection

11 [Type-Safe linking with recursive DLLs and shared libraries](#)



Dominic Duggan

November 2002 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 24 Issue 6

Publisher: ACM

Full text available: pdf(653.62 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 3, Downloads (12 Months): 42, Citation Count: 5

Component-based programming is an increasingly prevalent theme in software development, motivating the need for expressive and safe module interconnection languages. Dynamic linking is an important requirement for module interconnection languages, as ...

Keywords: Dynamic Linking, Module Interconnection Languages, Recursive Modules, Shared Libraries

12 [Practicing JUDO: Java under dynamic optimizations](#)



Michał Cierniak, Guei-Yuan Lueh, James M. Stichnoth

August 2000 PLDI '00: Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation

Publisher: ACM

Full text available: pdf(190.06 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 17, Downloads (12 Months): 75, Citation Count: 52

A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient implementation of Just-In-Time (JIT) compilation, exception handling, synchronization mechanism, and garbage collection (GC). These components are tightly coupled ...

13 [Recompilation for debugging support in a JIT-compiler](#)



Mustafa M. Tikir, Jeffrey K. Hollingsworth, Guei-Yuan Lueh

November 2002 PASTE '02: Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering

Publisher: ACM

Full text available: [pdf\(89.55 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 1, Downloads (12 Months): 21, Citation Count: 0

A static Java compiler converts Java source code into a verifiably secure and compact architecture-neutral intermediate format, called Java *byte codes*. The Java byte codes can be either interpreted by a Java Virtual Machine or translated into ...

Keyw ords: Java, Java virtual machine debugger interface, debug information, dynamic recompilation, field access watch, just-in-time compilation

14 [A proposal for a namespace facility in APL+WIN](#)



David E. Siegel

June 2001 ACM SIGAPL APL Quote Quad, Volume 31 Issue 4

Publisher: ACM

Full text available: [pdf\(141.83 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#)

Bibliometrics: Downloads (6 Weeks): 0, Downloads (12 Months): 4, Citation Count: 0

"Namespaces" have been implemented in some APL dialects, to reduce name scope and avoid name conflicts. Other implementations, including APL+WIN, do not yet support such a feature. This is an unofficial proposal for the design and implementation of such ...

15 [Testing Homotopy for paths in the plane](#)



Sergio Cabello, Yuanxin Liu, Andrea Mantler, Jack Snoeyink

June 2002 SCG '02: Proceedings of the eighteenth annual symposium on Computational geometry

Publisher: ACM

Full text available: [pdf\(236.53 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 4, Downloads (12 Months): 27, Citation Count: 5

In this paper we present an efficient algorithm to test if two given paths are homotopic; that is, whether they wind around obstacles in the plane in the same way. For simple paths specified by n line segments with obstacles described by n ...

Keyw ords: Homotopy, computational topology, simple paths

16 [Take Command](#)

Graydon L. Ekdahl

August 1998 Linux Journal, Volume 1998 Issue 52es

Publisher: Specialized Systems Consultants, Inc.

Full text available: [html\(9.64 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 5, Downloads (12 Months): 26, Citation Count: 0

Implementing a deltree Command in Linux: Removing a software package is made easyusing Dr. Ekdahl's deltree command

17 [Practicing JUDO: Java under dynamic optimizations](#)

Michał Cierniak, Guei-Yuan Lueh, James M. Stichnoth

May 2000 ACM SIGPLAN Notices, Volume 35 Issue 5

Publisher: ACM

Full text available: [pdf\(190.06 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 17, Downloads (12 Months): 75, Citation Count: 52

A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient implementation of Just-In-Time (JIT) compilation, exception handling, synchronization mechanism, and garbage collection (GC). These components are tightly coupled ...

18 [Addressing dynamic issues of program model checking](#)

Flavio Lerda, Willem Visser

May 2001 SPIN '01: Proceedings of the 8th international SPIN workshop on Model checking of software

Publisher: Springer-Verlag New York, Inc.

Full text available: [pdf\(213.42 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#)

Bibliometrics: Downloads (6 Weeks): 6, Downloads (12 Months): 32, Citation Count: 3

Model checking real programs has recently become an active research area. Programs however exhibit two characteristics that make model checking difficult: the complexity of their state and the dynamic nature of many programs. Here we address both these ...

19 [Distance browsing in spatial databases](#)

Gísli R. Hjaltason, Hanan Samet

June 1999 ACM Transactions on Database Systems (TODS), Volume 24 Issue 2

Publisher: ACM

Full text available: [pdf\(460.81 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 23, Downloads (12 Months): 203, Citation Count: 81

We compare two different techniques for browsing through a collection of spatial objects stored in an R-tree spatial data structure on the basis of their distances from an arbitrary spatial query object. The conventional approach is one that makes use ...

Keyw ords: R-trees, distance browsing, hierarchical spatial data structures, nearest neighbors, ranking

20 [Open runtime platform: flexibility with performance using interfaces](#)

Michał Cierniak, Brian T. Lewis, James M. Stichnoth

November 2002 JGI '02: Proceedings of the 2002 joint ACM-ISCOPE conference on Java Grande





Publisher: ACM

Full text available: [pdf\(300.99 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Bibliometrics: Downloads (6 Weeks): 10, Downloads (12 Months): 35, Citation Count: 3

According to conventional wisdom, interfaces provide flexibility at the cost of performance. Most high-performance Java virtual machines today tightly integrate their core virtual machines with their just-in-time compilers and garbage collectors to get ...

Keyw ords: JVM, Java, dynamic compilation, garbage collection, interface design, interfaces, just-in-time compilation, modular components, virtual machine

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)